

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(национальный исследовательский университет)

ФИЗТЕХ-ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра математического моделирования и прикладной математики

Квалификационная работа на соискание степени бакалавра
по направлению 03.03.01 «Прикладные математика и физика»,
профиль подготовки «Математическое моделирование,
вычислительная математика и физика»

Поиск оптимальных траекторий космического аппарата с использованием кривых, заполняющих пространство

Выполнил:

студент группы 771

Тарасов Андрей Алексеевич

Научный руководитель:

к.ф.-м.н., с.н.с

Трофимов Сергей Павлович

Москва, 2021

Аннотация

В работе рассматривается возможность применения методов глобальной оптимизации, которые используют заполняющие пространство кривые, к задачам динамики космического полета. Оптимизационные задачи, возникающие при проектировании космических миссий, зачастую существенно нелинейны и многоэкстремальны. Метод, который предлагается использовать, разработан для минимизации липшицевых функций, не требует гладкости функции и априорной информации о значении константы Липшица. Доказана его сходимости. В данной работе этот алгоритм реализован и применен к решению двух классических задач: задачи Ламберта и задачи оптимального перелета космического аппарата с двигателем ограниченной мощности. Реализована ускоренная модификация геометрического алгоритма одномерной оптимизации. Алгоритм распараллелен. При решении первой задачи алгоритм успешно завершается нахождением точки, в которой значения функционала достаточно мало, однако время работы слишком велико и ставится под вопрос сходимости к глобальному экстремуму. При решении второй задачи алгоритм не сходится. Тем самым, дается отрицательный ответ на вопрос о целесообразности использования рассматриваемого метода для решения оптимизационных задач динамики космического полета.

Содержание

Введение	3
1. Кривые Пеано	
и оптимизационные алгоритмы,	
использующие кривые Пеано	7
1.1. Определение кривой Пеано	7
1.2. Построение аппроксимации кривой Пеано	8
1.3. Аппроксимации кривых Пеано	12
1.4. Приложения к задачам глобальной оптимизации	17
1.5. Методы одномерной оптимизации	18
1.6. Параллельные алгоритмы	22
1.7. Реализованные алгоритмы	24
2. Задача Ламберта	26
2.1. Постановка задачи	26
2.2. Численные эксперименты	27
3. Задача оптимального перелета	
с квадратичным функционалом	30
3.1. Постановка задачи	30
3.2. Принцип максимума	32
3.3. Численные эксперименты	34
Список литературы	38

Введение

При проектировании миссий космических аппаратов (КА) возникает множество оптимизационных задач и задач оптимального управления. Их подробная классификация сама по себе представляет непростую задачу, пример такой классификации изложен в докладе Европейского космического агентства [1]. С математической точки зрения задачи различаются оптимизируемыми функционалами, граничными условиями, фиксированными переменными и переменными, по которым проводится оптимизация. От этого зависит сложность поставленной задачи.

Тип доступного управления зависит от двигателя, которым оснащен КА. Помимо химических двигателей реактивной тяги, которые позволяют считать мгновенным приращение скорости КА, широко распространены двигатели малой тяги, работу которых приходится моделировать непрерывными функциями от времени. Подробный обзор современных двигателей можно найти в [2].

К задачам первого класса с мгновенным приращением скорости относятся классическая задача Ламберта, задача о перелете между орбитами, гомановский перелет, многоимпульсные маневры.

Остановимся подробнее на задаче Ламберта, которая является одной из простейших и при этом важнейших задач при анализе космических миссий. В 1761 году Ламбертом была доказана теорема, которая утверждает, что время перелета по кеплеровской орбите между двумя заданными точками есть функция большой полуоси, суммы расстояний от притягивающего центра до начальной и конечной точек и длины хорды, соединяющей эти точки [3]. Существует несколько десятков методов решения задачи, большинство из которых выражают время перелета через описанные выше переменные или производные от них параметры и затем сводятся к итеративной процедуре. Например, свой метод предложил еще К. Гаусс,

взяв в качестве независимых переменных полуразность экстрентрических аномалий начальной и конечной точек и так называемый "sector-triangle ratio" [4]. Однако этот метод не сходится при угловой дальности больше π . Р. Баттин, используя аппарат гипергеометрических функций, модифицировал метод Гаусса, тем самым обеспечив сходимость и при больших угловых дальностях [4]. Позднее его метод был обобщен на случай многовитковых траекторий, то есть траекторий, при которых КА достигал конечной точки в нужный момент времени, перед этим пройдя через эту точку несколько раз [5]. Даже в последнее время появляются работы, посвященные этой задаче: Д. Иццо в 2014 году предложил свой алгоритм, сходящийся практически при любых условиях и при этом быстро, за единицы итераций [6].

К задачам второго класса с непрерывной тягой относятся задача оптимального перелета с квадратичным функционалом (с ограниченной мощностью двигателя малой тяги), задача минимизации времени перелета и задача минимизации топлива, затрачиваемого двигателем с постоянной скоростью истечения. Методы решения таких задач делятся на прямые и непрямые. Прямые методы дискретизируют все непрерывные функции времени и сводят задачу оптимального управления к задаче конечномерной оптимизации. Сложности применения прямых методов связаны с подбором дискретизации и выбором подходящего оптимизационного алгоритма для задачи в пространстве большой размерности. Непрямые методы чаще всего основаны на применении принципа максимума Понтрягина, которая затем решается какими-либо численными методами.

Одним из популярных методов решения краевой задачи для задачи оптимального перелета КА с двигателем ограниченной мощности является метод продолжения по параметру, разработанный В. Петуховым [7]. Оптимизационные задачи возможно решать с применением генетических или эволюционных алгоритмов [8]. Ранее упоминавшийся Д. Иццо разработал комплекс программ, позволяющий распараллеливать известные методы

оптимизации с помощью островной топологии [9]. Существуют и методы, основанные на подходящей замене переменных, например, переменные Кустаанхеймо-Штифеля позволяют избавиться от сингулярности уравнений движения в задаче двух тел посредством перехода в четырехмерное пространство из физического трехмерного [10].

Тем самым, подавляющее большинство задач может быть сведено к оптимизационной задаче, для которой необходимо подобрать подходящий алгоритм. Как правило, такие задачи существенно нелинейные, многоэкстремальные и негладкие. Разработано множество эффективных методов глобальной оптимизации [11], [8], которые традиционно тестируются на специальном классе задач. Однако некоторые исследователи считают [12], что такие задачи слишком просты и их успешное решение не позволяет гарантировать сходимость методов при решении практических задач, например, задач динамики космического полета. Существуют статьи [13], в которых описываются специально разработанные задачи для тестирования методов глобальной оптимизации.

Один из методов глобальной оптимизации был разработан нижегородским математиком Р. Стронгиным и подробно описан в монографии [14]. С помощью кривых, заполняющих пространство, многомерный функционал сводится к функции, определенной на отрезке числовой прямой, которая затем оптимизируется геометрическими или информационными алгоритмами. Для этого метода доказана теорема о сходимости к глобальному экстремуму, что выделяет его на фоне других методов глобальной оптимизации (например, генетических и эволюционных алгоритмов). Также Р. Стронгиным и его учениками были разработаны параллельные алгоритмы, что позволяет уменьшить время работы методов [15].

В данной работе анализируется возможность применения метода глобальной оптимизации, использующего кривые, заполняющие пространство, к двум характерным задачам, возникающим при планировании миссий

КА: задаче Ламберта и задаче оптимального перелета КА с двигателем ограниченной мощности.

По материалам данной работы сделан доклад «Оптимизация межпланетных перелетов с использованием кривых, заполняющих пространство» на секции динамики и управления движением космических аппаратов 63-й Всероссийской научной конференции МФТИ.

Текст структурирован следующим образом. В первой главе подробно изложен материал о кривых, заполняющих пространство, и оптимизационных алгоритмах, их использующих. После начальных определений описывается способ конструирования таких кривых и методы их аппроксимаций. Представлена процедура редукции многомерных задач к одномерным, и затем излагаются алгоритмы одномерной оптимизации. После этого обсуждаются возможности ускорения метода, в том числе с помощью распараллеливания. В заключении этой главы описывается реализованная на языке Python программа. Во второй главе дается постановка задачи Ламберта и описание численных экспериментов ее решения с помощью кривых, заполняющих пространство. В третьей главе дается постановка задачи оптимального перелета космического аппарата с двигателем ограниченной мощности. Описано сведение задачи оптимального перелета к краевой задаче с помощью принципа максимума Понтрягина, получающаяся краевая задача сведена к оптимизационной относительно нормы вектора невязки. Приведены результаты численных экспериментов. В заключении представлены выводы проведенного исследования.

1. Кривые Пеано

и оптимизационные алгоритмы, использующие кривые Пеано

В данной главе сначала обсудим кривые, заполняющие пространство (также в дальнейшем называемые кривыми Пеано), с абстрактной математической точки зрения и способы их построения, затем их аппроксимации и возможность приложения к задачам глобальной оптимизации. Первые 7 рисунков в этой главе заимствованы из [17], последние два – из [18].

Кривые, заполняющие пространство, активно используются в разных дисциплинах, таких как: компьютерное моделирование, вычислительная гидродинамика и геофизика, распознавание и обработка изображений, анализ графов [16]. Алгоритмы, основанные на кривых Пеано, часто допускают распараллеливание, что и используется в данной работе.

1.1. Определение кривой Пеано

Условимся понимать под гиперкубом D в пространстве размерности N следующее множество точек:

$$D = \{y \in \mathbb{R}^N \mid -\frac{1}{2} \leq y_i \leq \frac{1}{2}, 1 \leq i \leq N\}.$$

Вопрос: существует ли непрерывное отображение отрезка прямой $[0, 1]$ на единичный гиперкуб D ?

Ответ: да. Впервые это показал в 1890 году знаменитый итальянский математик Дж. Пеано для случая $N = 2$ [19]. В 1891 году Д. Гильберт предоставил явный геометрический алгоритм построения таких отображений [20].

Тем самым, можно ввести следующее определение.

Определение 1.1. *Непрерывное отображение $y(x)$, $x \in [0, 1]$, на единичный гиперкуб D , называется кривой Пеано или кривой, заполняющей пространство.*

Сразу отметим, что данное отображение с необходимостью не инъективно. Важность этого факта станет прояснена в дальнейшем при мотивировке использования параллельных алгоритмов.

1.2. Построение аппроксимации кривой Пеано

Существует несколько способов построения кривых Пеано как объектов, возникающих при предельном переходе. Свои алгоритмы предложили многие математики, такие как Пеано, Гильберт, Мур, Лебег, Серпинский, Поля. Эти алгоритмы подробно изучены в монографии Сагана [21]. Изложим построение кривой, заполняющей пространство, следуя Гильберту, а затем обсудим различия кривых Гильберта и других.

Построение кривой Гильберта

Разделим единичный гиперкуб D на 2^N равных меньших гиперкубов, пересекая исходный гиперкуб N плоскостями, каждая из которых перпендикулярна одной из координатных осей и проходит через начало координат. Тем самым, получено 2^N меньших гиперкубов с длиной ребра 2^{-1} , каждый из которых обозначим как $D(z_1)$, где $0 \leq z_1 \leq 2^N - 1$. (Иллюстрации приведены для плоского случая: $N = 2$, рис. 1, 2). Каждый из полученных гиперкубов разделим подобным же образом. В результате, имеем 2^{2N} гиперкубов с длиной ребра 2^{-2} , которые можно занумеровать как $D(z_1, z_2)$, где $0 \leq z_1 \leq 2^N - 1, 0 \leq z_2 \leq 2^N - 1$. Продолжая разбиения аналогичным образом M раз, получаем 2^{MN} гиперкубов с длиной ребра 2^{-MN} , занумерованных таким образом, что выполняется следующее вложение

$$D \supset D(z_1) \supset D(z_1, z_2) \supset \dots \supset D(z_1, \dots, z_M),$$

где $0 \leq z_i \leq 2^N - 1, 1 \leq i \leq M$.

Теперь разделим единичный отрезок $[0, 1]$ на 2^N равных меньших отрезков и обозначим их как $d(z_1)$, где $0 \leq z_1 \leq 2^N - 1$. Каждый из полученных отрезков еще раз поделим на 2^N частей, которые обозначим как $d(z_1, z_2)$, где $0 \leq z_1 \leq 2^N - 1, 0 \leq z_2 \leq 2^N - 1$. После M итераций получаем разбиение единичного отрезка на 2^{MN} меньших, каждый длиной 2^{-MN} , для которых выполняется следующее вложение

$$d \supset d(z_1) \supset d(z_1, z_2) \supset \dots \supset d(z_1, \dots, z_M),$$

где $0 \leq z_i \leq 2^N - 1, 1 \leq i \leq M$.

Процесс деления единичного отрезка можно воспринимать как описанный выше процесс деления гиперкуба в пространстве с $N = 1$ – с той лишь разницей, что левый конец отрезка перенесен в начало координат. Переобозначим отрезок $d(z_1, \dots, z_M)$ с левым концом ν как $d(M, \nu)$.

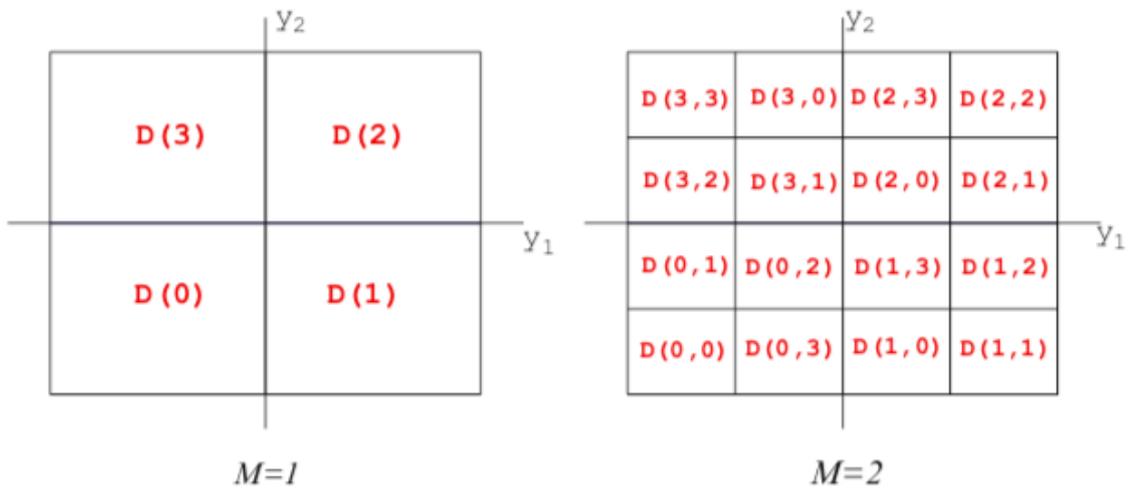


Рисунок 1 – Пример разбиения гиперкуба.

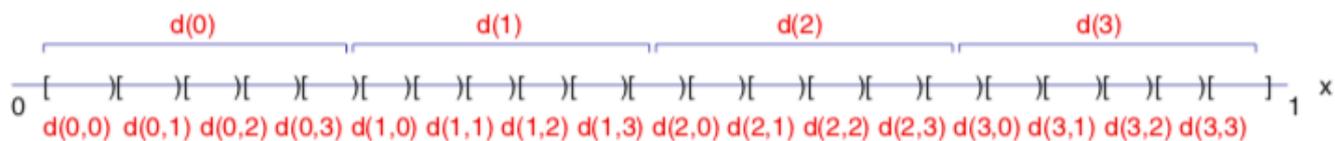


Рисунок 2 – Пример разбиения отрезка.

Установим взаимнооднозначное соответствие между 2^{MN} отрезками и 2^{MN} гиперкубами, то есть установим отображение $d(M, \nu) \rightarrow D(M, \nu)$ так, чтобы оно удовлетворяло следующим свойствам:

1. $D(M + 1, \nu') \subset D(M, \nu'')$, если и только если $d(M + 1, \nu') \subset d(M, \nu'')$.
2. Два отрезка $d(M, \nu')$, $d(M, \nu'')$ имеют общую граничную точку, если и только если соответствующие гиперкубы $D(M, \nu')$, $D(M, \nu'')$ имеют общую грань.

Первое условие выполнено по построению, выполнение второго достигается специальной нумерацией гиперкубов. Опустим алгоритм данной нумерации, который подробно описан в монографии Р. Стронгина [14], лишь отметив, что полученное взаимнооднозначное соответствие обладает важными свойствами, сформулированными в следующем утверждении.

Теорема 1.1. Пусть $y(x)$ – отображение, определенное следующим правилом: для любого $M \geq 1$ $y(x) \in D(M, \nu)$ тогда и только тогда, когда $x \in d(M, \nu)$. Тогда:

1. $y(x)$ – непрерывное отображение единичного отрезка $[0, 1]$ на гиперкуб D , то есть полученное отображение действительно является кривой, заполняющей пространство.
2. Рассмотрим Липшицеву функцию $F : D \rightarrow \mathbb{R}$, с константой Липшица L , то есть такую функцию, разность значений которой в

двух точках можно ограничить расстоянием между этими двумя точками, с точностью до конечного множителя: $\forall y_1, y_2 \in D \quad |F(y_1) - F(y_2)| < L\|y_1 - y_2\|$. Тогда функция одной переменной $f(x) = F(y(x)) : [0, 1] \rightarrow \mathbb{R}$ является гёльдеровой с показателем N^{-1} и константой Гёльдера $H = 2L\sqrt{N+3}$. То есть $f(x)$ удовлетворяет соотношению

$$\forall x_1, x_2 \in [0, 1] \quad |f(x_1) - f(x_2)| < 2L\sqrt{N+3}|x_1 - x_2|^{1/N}.$$

Еще раз повторимся, что кривая, заполняющее пространство, есть объект, получаемый при предельном переходе, а потому сами по себе такие кривые не представляют практического интереса: с такими объектами просто нет возможности работать. В следующем параграфе будут обсуждаться способы аппроксимации кривых Пеано.

Выше был изложен алгоритм, предложенный Гильбертом. Такой способ построения обуславливает значение константы Гёльдера $H = 2L\sqrt{N+3}$. Так как в дальнейшем функционал $F(y(x))$ будет оптимизироваться, хотелось бы подобрать отображение единичного отрезка на гиперкуб таким образом, чтобы значение константы Гельдера было минимальным и функционал был более "плавным". Для формального обсуждения этого вопроса в пространстве размерности $N = 2$ введем

Определение 1.2. Для кривой Пеано $y : [0, 1] \rightarrow [-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$ квадратно-линейным отношением называется следующая величина:

$$\sup \frac{|y(x_1) - y(x_2)|^2}{|x_1 - x_2|},$$

где супремум берется по всем парам точек из единичного отрезка.

Квадратно-линейное отношение определяет значение константы Гильберта. Данное определение легко обобщается на случай произвольной раз-

мерности пространства, однако пока не получено никаких существенных результатов для $N > 2$.

Российский математик К. Бауман доказал, что квадратно-линейное отношение для обсуждавшейся выше кривой Гильберта равно 6 (то есть константа Гёльдера $H = 2\sqrt{3}$, а не $2\sqrt{5}$, как было показано простыми оценками при доказательстве теоремы 1.1) [22]. Им же была найдена кривая с меньшим квадратно-линейным-отношением, равным $5\frac{1}{3}$ [23]. Это кривая фрактального рода 9, то есть итеративный процесс происходит за счет деления квадрата не на 4 части, а на 9. Первый пример кривой, заполняющей пространство, который принадлежал самому Пеано – это тоже кривая фрактального рода 9. При этом известна оценка снизу для квадратно-линейного отношения произвольной кривой ПеаноЖ эта оценка равна 4.

Вопросы нахождения кривых с минимальным квадратно-линейным отношением (или с обобщением этого определения для пространств большей размерности) сложны и представляют скорее теоретический интерес. Сомнительно, что незначительное уменьшение константы Гёльдера сильно ускорит оптимизационный алгоритм.

1.3. Аппроксимации кривых Пеано

Следуя монографии Р. Стронгина [14], опишем построение аппроксимации кривых Пеано.

Фиксируем число разбиения M . Рассмотрим кусочек разбиения единичного отрезка

$$d(z_1, \dots, z_M) = [v_i, v_i + 2^{-MN}), \quad 0 \leq i \leq 2^{MN} - 1.$$

Выше было сказано, что существует специальное отображение таких отрезков на гиперкубы разбиения $D(z_1, \dots, z_M)$. Условимся обозначать центр такого гиперкуба, как

$$y^i = y(z_1, \dots, z_M), \quad 0 \leq i \leq 2^{MN} - 1.$$

Указанное отображение из единичного отрезка на центры кубов уже можно считать аппроксимацией кривой Пеано, которое, однако, не является непрерывным. Построим непрерывную кривую $l(x)$, соединив эти центры.

$$l(x) = y^i + (y^{i+1} - y^i) [(w(x) - v_i) / (v_{i+1} - v_i)]$$

$$v_i \leq w(x) \leq v_{i+1} \text{ и } w(x) = x(1 - 2^{-MN}), \quad 0 \leq x \leq 1.$$

Образ следующего отрезка на прямой

$$\left[v_i (1 - 2^{-MN})^{-1}, v_{i+1} (1 - 2^{-MN})^{-1} \right], \quad 0 \leq i < 2^{MN} - 1$$

есть отрезок, соединяющий центры гиперкубов. Поэтому полученная аппроксимация называется кусочно-линейной. Такое отображение является непрерывным и инъективным. Иллюстрация для плоского случая с $M = 3$ приведена рис. 3.

Истинная кривая Пеано, как было сформулировано в теореме 1.1, из многомерной липшицевой функции делает гёльдерову. Аналогичное утверждение справедливо и для аппроксимации.

Теорема 1.2. *Если функция $F(y) : F : D \rightarrow \mathbb{R}$ удовлетворяет условию Липшица с константой L , то суперпозиция $F(l(x)) : [0, 1] \rightarrow \mathbb{R}$ является гёльдеровой с константой $H = 2L\sqrt{N + 3}$.*

Приведенный способ соединения центров гиперкубов, очевидно, не единственный, однако именно он делает справедливым утверждение только что сформулированной теоремы. Если рассмотреть развертки, изображенные на рис. 4, то суперпозиция липшицевой функции и такой кривой уже не будет удовлетворять условию Гёльдера.

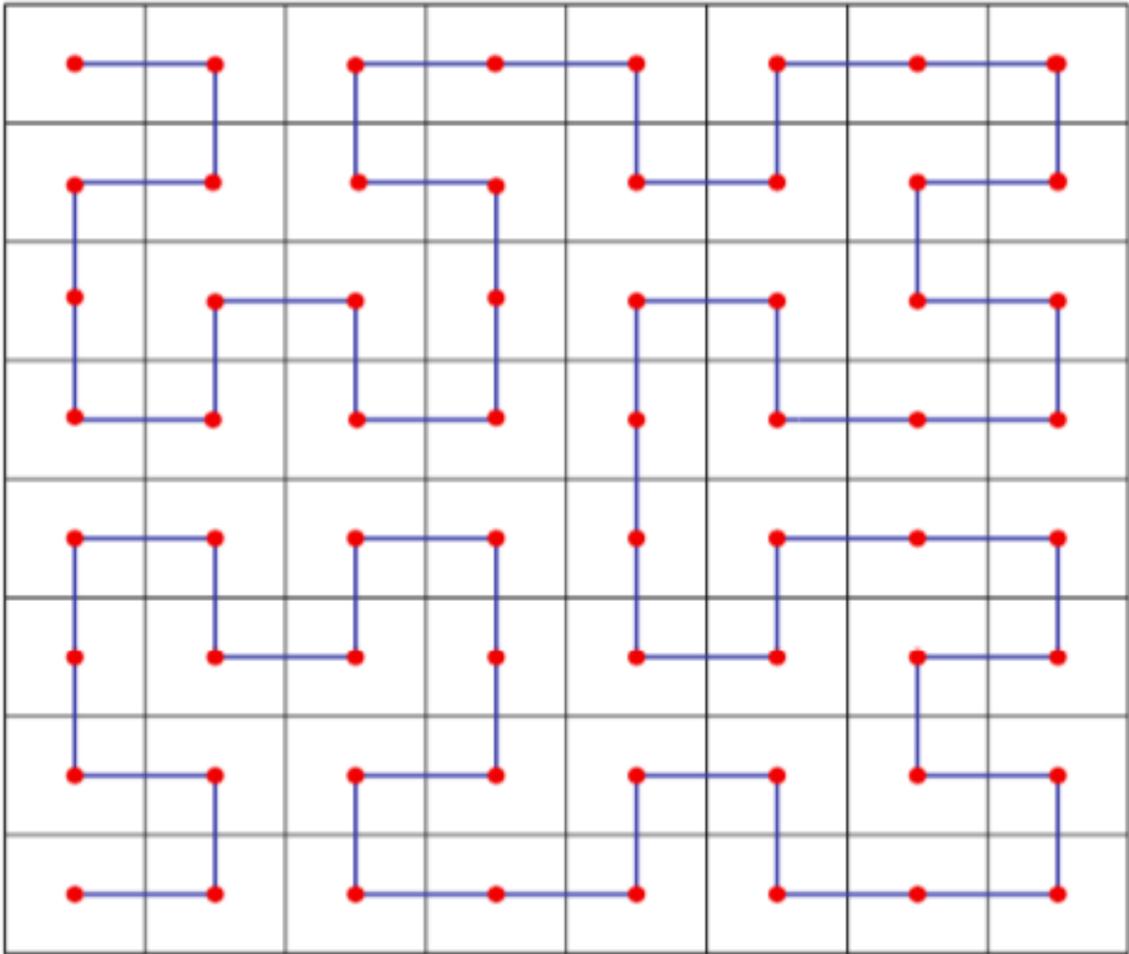


Рисунок 3 – Кусочно-линейная аппроксимация.

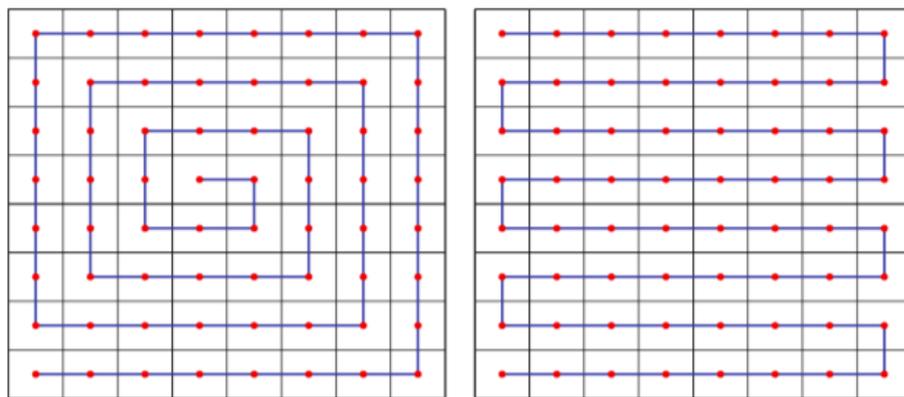


Рисунок 4 – Кусочно-линейные аппроксимации, не удовлетворяющие теореме.

Подобрав достаточно большое число разбиения M , можно добиться достаточной точности приближения точек гиперкуба точками кусочно-

линейной кривой. Однако у этой аппроксимации есть два недостатка.

1. Образ кусочно-линейной аппроксимации с числом разбиения M не обязан принадлежать образу кусочно-линейной аппроксимации с большим числом разбиения $M' > M$. Это может оказаться существенным, если в процессе работы определенного алгоритма текущее значение M окажется недостаточным и его необходимо будет увеличить. Тогда уже вычисленные значения функционала могут остаться неиспользуемыми.
2. Вторым недостатком связан с уже упоминавшейся неинъективностью кривой Пеано, в то время как $l(x)$ инъективна. Гиперкуб разбиения в пространстве размерности N имеет вплоть до 2^N "соседей" (при предельном переходе это свойство выливается в факт, что каждая точка в гиперкубе имеет вплоть до 2^N прообразов), однако кусочно-линейная аппроксимация уменьшает количество "соседей" до двух: слева и справа (рис. 5). Так как практически все оптимизационные алгоритмы используют близость между точками, инъективность аппроксимации означает потерю существенной информации.

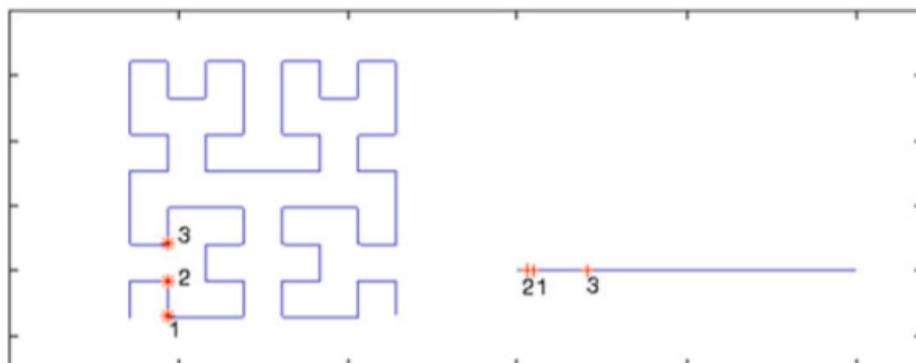


Рисунок 5 – "Соседи" на единичном квадрате и его развёртке.

Одним из способов устранения этих недостатков является конструирование новой аппроксимации. Пусть $P(M, N)$ равномерная сетка на кубе D ,

расстояние между узлами сетки равно 2^{-M} , всего таких узлов 2^{MN} . Отметим, что эта сетка равномерная, то есть узлы такой сетки приходятся на вершины гиперкубов разбиения, а не на центры этих гиперкубов. Равномерно разобьем единичный отрезок на 2^{MN} точек. Существует специальное отображение точек отрезка на $P(M, N)$, которое является кусочно-постоянным [17]. Иллюстрация неинъективной аппроксимации Пеано в случае $N = 2, M = 3$ предоставлена на рис. 6. Красными точками обозначены узлы решетки, а натуральные числа возле них нумеруют последовательные точки из единичного отрезка.

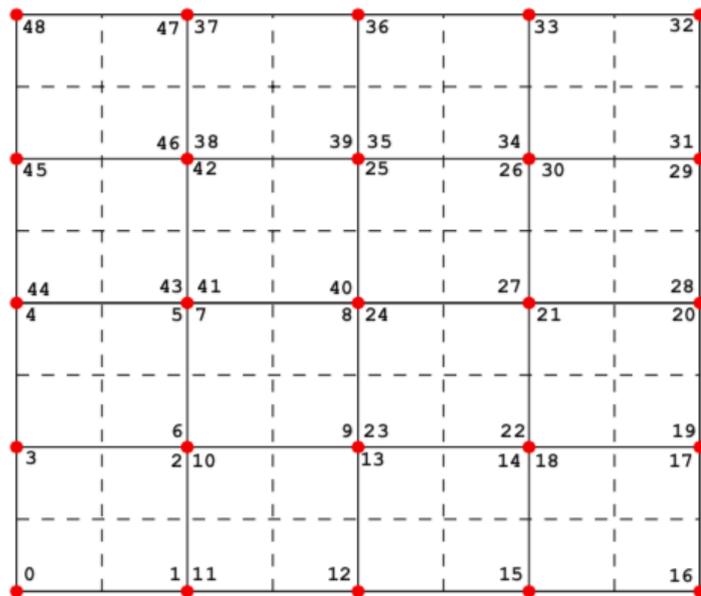


Рисунок 6 – Неинъективная развертка

Видно, что у такой аппроксимации нет описанных выше недостатков. Однако в данной работе используется кусочно-линейная аппроксимация. Значение числа M изначально выбирается максимально возможным, а потому первый недостаток себя не проявляет. Вторым недостатком устраняется использованием множественных разверток, что будет подробно описано ниже при обсуждении распараллеливания оптимизационного алгоритма.

1.4. Приложения к задачам глобальной оптимизации

Обсудим приложения кривых, заполняющих пространство, к задачам оптимизации.

Рассмотрим функционал $F(y)$, определенный на гиперкубе D . Предполагается, что функционал многоэкстремален и удовлетворяет условию Липшица. Дифференцируемость или гладкость не предполагается. Требуется найти глобальный минимум. $l(x)$ – кусочно-линейная аппроксимация кривой Пеано для фиксированного M . Рассмотрим функцию

$$f(x) = F(l(x)) : [0, 1] \rightarrow \mathbb{R}.$$

Теорема 3.2 утверждает, что такая функция удовлетворяет условию Гёльдера с показателем $\alpha = N^{-1}$ и константой Гёльдера $H = 2L\sqrt{N+3}$, то есть

$$\forall x_1, x_2 \in [0, 1] \quad |f(x_1) - f(x_2)| < H|x_1 - x_2|^{1/N}.$$

Имеет место следующая теорема, доказанная в [14].

Теорема 1.3. Пусть U_M^* – нижняя оценка функции $F(y)$ вдоль кусочно-линейной аппроксимации кривой Пеано $l(x)$, то есть имеет место неравенство

$$U_M^* \leq F(l(x)), \quad x \in [0, 1].$$

Тогда $U = U_M^* - 2^{-(M+1)}L\sqrt{N}$ является нижней оценкой функции $F(y)$, то есть

$$U \leq F(y), \quad y \in D.$$

Тем самым, с помощью кривых, заполняющих пространство, задача многомерной оптимизации сведена к задаче оптимизации одномерной гёльдеровской функции. Обсудим методы одномерной оптимизации.

1.5. Методы одномерной оптимизации

Как было сформулировано выше, оптимизируемый функционал предполагается гёльдеровым. Однако сначала изложим алгоритм Пиявского [24] для липшицевых функций, так как идеи, лежащие в его основе, важны для дальнейших методов.

Алгоритм Пиявского

Рассматривается липшицева функция $f(x)$, $x \in [a, b]$, значение которой в точке минимума x^* обозначим как $f^* = f(x^*)$. Условие Липшица можно переформулировать следующим образом: для произвольного $y \in [a, b]$ функция $C(x) = f(y) - L|x - y|$ ограничивает снизу $f(x)$, то есть $C(x) \leq f(x)$, $x \in [a, b]$.

Пусть известны значения функции в k последовательных точках: x_1, \dots, x_k из отрезка $[a, b]$ и z_1, \dots, z_k — значения функции в этих точках. Сконструируем кусочно-линейную функцию:

$$C^k(x) = \cup_{i=2}^k c_i(x),$$

$$c_i(x) = \max\{z_{i-1} - L(x - x_{i-1}), z_i + L(x - x_i)\}, \quad x \in [x_{i-1}, x_i].$$

Такая функция, в силу условия Липшица, ограничивает $f(x)$ снизу: в дальнейшем будем называть ее минорантой. Миноранта построена как поточечный максимум линейных функций, ограничивающих функцию $f(x)$ снизу на отрезке $[x_{i-1}, x_i]$. Точки пересечения этих линейных функций вычисляются как:

$$\tilde{x}_i = \frac{x_i + x_{i-1}}{2} + \frac{z_{i-1} - z_i}{2L},$$

а значения миноранты в этих точках равны

$$c_i(\tilde{x}_i) = \frac{z_i + z_{i-1}}{2} + L \frac{x_{i-1} - x_i}{2}.$$

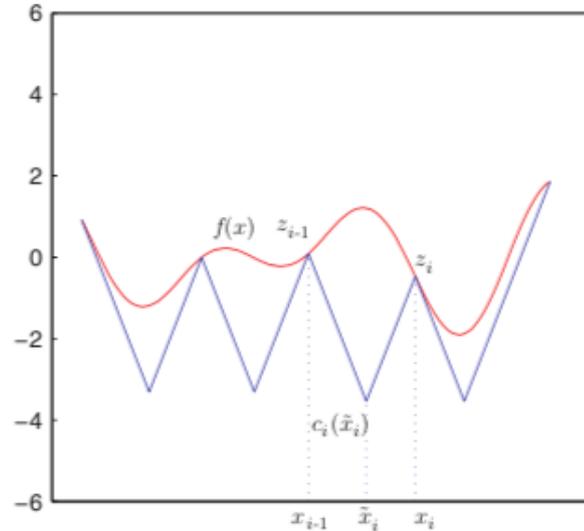


Рисунок 7 – Пример кусочно-линейной миноранты.

На рис. 7 приведен пример построения кусочно-линейной миноранты. Алгоритм Пиявского заключается в выборе минимальной точки миноранты как следующей для вычисления значения функции. То есть точка x^{k+1} выбирается таким образом, что

$$C^k(x^{k+1}) = \min C^k(x), \quad x \in [a, b]$$

или

$$C^k(x^{k+1}) = \min c_i(\tilde{x}_i), \quad 1 \leq i \leq k.$$

После K шагов $f_K^* = \min\{z_i, 1 \leq i \leq K\}$ считается оценкой минимума функции и $x_K^* = \arg \min\{z_i, 1 \leq i \leq K\}$ – оценкой точки минимума.

Доказана сходимость алгоритма Пиявского, если за L принимается число, не меньшее истинного значения константы Липшица. Однако не всегда возможно найти истинное значение константы. Чтобы преодолеть эту трудность, Р. Стронгин предложил алгоритм, адаптивно оценивающий константу [14]. В предположении, что разность значений функций есть нормально распределенная случайная величина, зависящая от расстояния между аргументами функции, на множестве вычисленных значений вводится вероятностное пространство и на каждой итерации алгоритм выбирает

точку, где с наибольшей вероятностью находится минимум.

Геометрический алгоритм

Теперь рассмотрим алгоритм для оптимизации одномерной Гёльдеровой функции с показателем $\alpha = 1/N$. Аналогично алгоритму Пиявского этот метод строит миноранту функции и на каждой итерации выбирает точку, доставляющую минимум миноранте [17]. Константу Гёльдера H также возможно оценивать адаптивно.

Алгоритм работает следующим образом. Сначала вычисляются значения на концах отрезка $x^1 = a$, $x^2 = b$. После k итераций выбор точки для следующей итерации x^{k+1} осуществляется по следующим правилам:

1. Перенумеруем точки x^1, x^2, \dots, x^k в процессе возрастания, используя нижние индексы, то есть $a = x_1 < x_2 < \dots < x_k = b$ и обозначим $z_i = f(x_i)$, $1 \leq i \leq k$.
2. Примем следующую оценку константы Гельдера на текущем шагу:

$$h^k = \max \{h_i : 2 \leq i \leq k\}$$

$$h_i = \frac{|z_i - z_{i-1}|}{|x_i - x_{i-1}|^{1/N}}, \quad 2 \leq i \leq k$$

3. Для каждого интервала вычислим следующие величины. (r - заранее задаваемый параметр алгоритма):

$$y_i = \frac{1}{2} (x_i + x_{i-1}) - \frac{z_i - z_{i-1}}{2rm_i (x_i - x_{i-1})^{\frac{1-N}{N}}},$$

$$R_i = \min \left\{ f(x_{i-1}) - rm_i (y_i - x_{i-1})^{1/N}, f(x_i) - rm_i (x_i - y_i)^{1/N} \right\}.$$

4. Найдем интервал с минимальной характеристикой R_i

$$t = \arg \min \{R_i : 2 \leq i \leq k\}.$$

5. Если $x_t - x_{t-1} > \varepsilon$, то положим $x^{k+1} = y_t$. Иначе алгоритм считается закончившим свою работу с оцениваемым минимальным значением

$$f^* = \min\{z_i, 1 \leq i \leq k\}.$$

Данный алгоритм строит миноранту из линейных аппроксимаций гипербол, подпирающих минимизируемую функцию. Характеристика R есть ордината точки пересечения этих прямых. Существует алгоритм GJE, где характеристика вычисляется непосредственно как ордината точки пересечения самих гипербол, а константа Гёльдера не оценивается в процессе работы алгоритма [25]. При таком подходе на каждой итерации численно решается нелинейное уравнение.

Доказана теорема сходимости для геометрического алгоритма.

Теорема 1.4. Пусть x^* – точка глобального минимума $f(x)$. Тогда, если существует номер итерации k^* такой, что для всех $k > k^*$ выполняется неравенство

$$rm_{j(k)} > H_{j(k)},$$

где $H_{j(k)}$ – константа Гёльдера на отрезке $[x_{j(k)-1}, x_{j(k)}]$, то есть

$$|f(x) - f(y)| \leq H_{j(k)}|x - y|^{1/N}, \quad x, y \in [x_{j(k)-1}, x_{j(k)}],$$

и $x^* \in [x_{j(k)-1}, x_{j(k)}]$, тогда последовательность $\{x^k\}$ сходится к x^* .

Ускорение геометрического алгоритма

Геометрический алгоритм использует глобальную оценку константы Гёльдера, которая на деле может сильно отличаться в различных участках единичного отрезка. Существует модификация [17], которая для каждого отрезка $[x_{i-1}, x_i]$ выбирает оценку константы Гёльдера как некоторую комбинацию глобальной и локальной оценки константы. Эта модифика-

ция позволяет существенно ускорить алгоритм, сохранив справедливыми утверждения о его сходимости.

Для каждого отрезка $[x_{i-1}, x_i]$ вычислим следующие величины:

$$\begin{aligned}\lambda_i &= \max \{h_{i-1}, h_i, h_{i+1}\}, \\ h_i &= \frac{|z_i - z_{i-1}|}{|x_i - x_{i-1}|^{1/N}}, \quad 3 \leq i \leq k-1, \\ \gamma_i &= h^k \frac{x_i - x_{i-1}}{X^{\max}}, \\ h^k &= \max \{h_i\},\end{aligned}$$

$$X^{\max} = \max \{x_i - x_{i-1} : 2 \leq i \leq k\}.$$

Тогда для каждого отрезка предлагается использовать следующую оценку константы Гельдера:

$$m_i = \max \{\xi, \lambda_i, \gamma_i\},$$

где ξ – малый параметр алгоритма.

1.6. Параллельные алгоритмы

Кусочно-линейная аппроксимация кривой Пеано позволяет свести многомерную оптимизационную задачу к задаче оптимизации одномерной гёльдеровской функции. Как уже было отмечено выше, такая аппроксимация является инъективной, в отличие от самой кривой Пеано. Это означает, что близким точкам на гиперкубе соответствуют далекие точки на единичном отрезке.

Одно из решений этой проблемы заключается в использовании множественных разверток, что позволяет сохранить информацию о близости точек [18]. Оказывается справедливым утверждение, что найдется хотя бы одна развертка такая, что паре близких точек из гиперкуба будут

соответствовать близкие точки на единичном отрезке. Задается не одно отображение $l_M(x)$, а несколько отображений, получаемых поворотом относительно центра гиперкуба на $\pm\pi/2$. Поворот осуществляется в плоскости, параллельной двум координатным осям.

На рис. 8 в плоском случае изображены две развертки, одна из которых повернута относительно второй на $\pi/2$. Стрелкой указаны две близкие точки, которые имеют далекие прообразы в случае одной кривой Пеано, и близкие в другом.

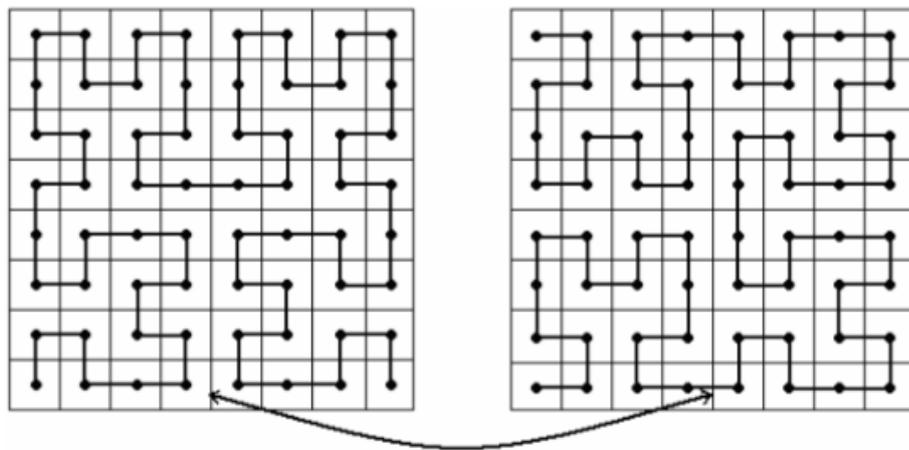


Рисунок 8 – Две развертки с разными свойствами близости прообразов двух выбранных точек.

Поворот развертки реализуется с помощью матрицы. В качестве примера рассмотрим поворот в плоскости, параллельной 2-й и 4-й оси в пространстве размерности $N = 5$. Матрица поворота получается из единичной заменой второй и четвертой строк и столбцов (см. рис. 9).

Всего таких поворотов в пространстве размерности N может быть получено $N(N - 1)$. Конечно, можно поворачивать на угол π или осуществлять другие преобразования.

Опишем параллельный алгоритм оптимизации многомерной липшецевой функции, использующий множественные развертки. Пусть доступно запустить решение задачи на S параллельных процессах. Каждому процессу сопоставлена своя развертка, и каждый процесс решает свою одномерную

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Рисунок 9 – Матрицы поворота на углы $\pi/2$ и $-\pi/2$ вокруг нормали к плоскости, содержащей вторую и четвёртую координатные оси.

задачу геометрическим алгоритмом. После выполнения итерации процесс отправляет другим процессам точку из гиперкуба и вычисленное в ней значение. Другие процессы находят прообраз принятой точки и добавляют пару $(x, f(x))$ к множеству обработанных точек.

Отметим, что существует множество других подходов к распараллеливанию алгоритма. Наряду с вращающимися развертками, описанными выше, используются сдвиговые: сдвиг осуществляется вдоль главной диагонали. Существует простейший алгоритм, в котором S процессов совместно решают одномерную задачу: главный компьютер выбирает S отрезков с экстремальными характеристиками и распределяет их между процессами. Большое количество алгоритмов описано в монографии Р. Стронгина и Я. Сергеева [15].

1.7. Реализованные алгоритмы

Для решения задач динамики космического полета был написан блокнот Jupyter Notebook на языке программирования Python, где были реализованы оптимизационные алгоритмы и функции для работы с кривыми Пеано. Все элементы программы, кроме одного файла (см. пункт 2), написаны в блокноте.

Опишем проделанные этапы:

1. К блокноту с помощью модуля `ctypes` подключены файлы, написанные на языке Си. Функции, реализованные в этих файлах позволяют строить аппроксимации кривых Пеано – как кусочно-линейную, так и не инъективную. Код был написан В. Гергелем и опубликован, например, в работе [17]. Код был протестирован, ошибок не обнаружено.
2. На языке Си была реализована функция, возвращающая прообраз кусочно-линейной аппроксимации Пеано, и так же подключена к блокноту. При тестировании максимальная ошибка нахождения прообраза имеет порядок 10^{-16} . Считаем, что это ошибка порядка машинного нуля и функция работает корректно.
3. Реализованы геометрический и информационный алгоритмы оптимизации для одномерных функций. Алгоритмы протестированы на типовых задачах. Результаты (число итераций до сходимости) аналогичны приведенным в литературе [17].
4. Реализована модификация геометрического метода с ускорением (описан в разделе 3.5), метод протестирован на функциях Розенброка, Растрьгина и сферической. Результаты аналогичны приведенным в литературе.
5. С помощью модуля `multiprocessing` стандартной библиотеки Python реализована параллельная версия геометрического алгоритма с вращающимися развертками. Тестирование также не позволяет выявить ошибки работы программы.

Реализованный параллельный алгоритм применялся при решении двух оптимизационных задач астродинамики, о чем будет подробно рассказано в следующих главах.

2. Задача Ламберта

Глава состоит из двух разделов. В первом приводится постановка задачи, во втором описываются проведенные численные эксперименты и обсуждаются их результаты.

2.1. Постановка задачи

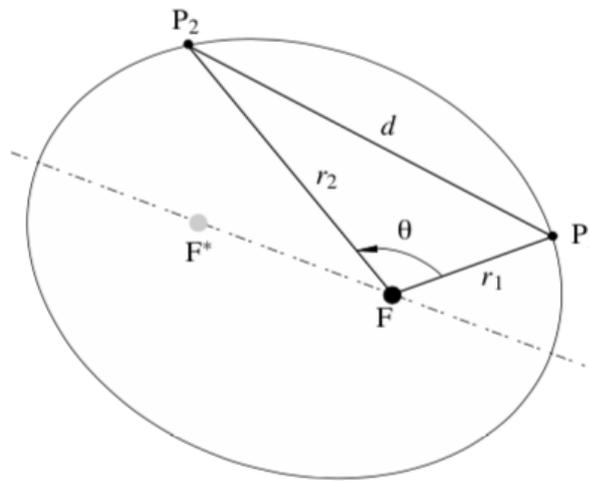


Рисунок 10 – Задача Ламберта [5].

Задача Ламберта ставится следующим образом. Заданы две точки в пространстве: P_1 и P_2 . Необходимо найти кеплерову орбиту, по которой КА сможет переместиться из точки P_1 в точку P_2 за заданное время T .

Согласно теореме Ламберта, время полета КА по кеплеровой орбите между заданными точками есть функция от трех переменных: большой полуоси эллипса (или гиперболы), суммы расстояний от точек до притягивающего центра и расстояния между точками:

$$T = T(a, r_1 + r_2, d) \quad (1)$$

Большинство существующих методов решения задачи Ламберта, как было отмечено во введении, выражают время полета через эти или другие

переменные и численно решают полученное уравнение.

Однако задачу Ламберта можно сформулировать как краевую или оптимизационную задачу.

Уравнения движения имеют вид

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v}, \\ \dot{\mathbf{v}} = -\frac{\mu}{r^3}\mathbf{r}. \end{cases} \quad (2)$$

При ограничениях:

$$\begin{cases} \mathbf{r}(t_0) = \mathbf{r}_0, \\ \mathbf{r}(t_0 + T) = \mathbf{r}_f \end{cases} \quad (3)$$

необходимо найти скорость в начальный момент времени $\mathbf{v}_0 = \mathbf{v}(t_0)$.

Введем функцию невязки \mathbf{f} как

$$\mathbf{f} = \mathbf{r}(t_0 + T) - \mathbf{r}_f. \quad (4)$$

Тем самым, получаем оптимизационную задачу

$$\mathbf{f}^2 \rightarrow \min, \quad (5)$$

где функция \mathbf{f} понимается как функция от вектора скорости в начальной момент времени.

2.2. Численные эксперименты

Рассматривалась плоская задача. В качестве начальной точки был выбран перигелий земной орбиты, в качестве конечной – афелий орбиты Марса. Использовались обезразмеренные переменные. Время полета было взято равным 499 дням. В качестве алгоритма интегрирования уравнений движения использовался встроенный в модуль `scipy` метод Рунге-Кутты

четвертого порядка ($\text{atol} = 10^{-10}$, $\text{rtol} = 10^{-10}$). Вычисления проводились на компьютере Intel Xeon CPU E3-1275 v5 @ 3.60GHz, имеющим 4 ядра, что позволяет распараллелить на максимум 8 потоков.

Было проведено четыре серии экспериментов. Сначала использовался параллельный геометрический алгоритм с тремя развертками, с уровнем доверия $r = 2$. Были получены решения для разных критериев остановки алгоритма, когда ε менялся в пределах от 10^{-3} до 10^{-6} . Результаты представлены в таблице 1. В качестве числа итераций бралось наибольшее среди трех процессов количество итераций до завершения работы алгоритма, m – максимальная среди трех процессов оценка константы Гёльдера, время округленно до целого числа минут. При $\varepsilon = 10^{-6}$ алгоритм не сошелся за 100 000 итераций.

Таблица 1 – Результаты численного эксперимента при $r = 2$. Основной алгоритм.

Точность	1e-3	1e-4	1e-5	1e-6
Число итераций	686	5 001	21 001	>100 000
Чначение невязки	2.8e-3	1.2e-4	1.5e-5	9.5e-7
Оценка константы Гёльдера m	512	1 947	3 463	45 699
Время (мин.)	1	5	21	79

Затем эксперимент был повторен с использованием ускоренной модификации параллельного геометрического алгоритма. ε менялся уже в пределах от 10^{-7} до 10^{-10} . Результаты приведены в таблице 2.

Таблица 2 – Результаты численного эксперимента при $r = 2$. Ускоренный алгоритм.

Точность	1e-7	1e-8	1e-9	1e-10
Число итераций	5 855	6 640	7 076	5 810
Чначение невязки	4.9e-6	1.5e-6	1.3e-7	3.9e-7
Оценка константы Гёльдера m	298	238	230	2 368 980
Время (мин.)	5	5	4	4

Аналогичные результаты проведены при $r = 5$. (Таблицы 3, 4.)

Таблица 3 – Результаты численного эксперимента при $r = 5$. Основной алгоритм.

Точность	1e-3	1e-4	1e-5	1e-6
Число итераций	690	4 213	16 879	>100 000
Значение невязки	9.7e-3	1.4e-3	1.2e-4	7.3e-6
Оценка константы Гёльдера m	198	247	343	2 496
Время (мин.)	1	4	14	76

Таблица 4 – Результаты численного эксперимента при $r = 5$. Ускоренный алгоритм.

Точность	1e-7	1e-8	1e-9	1e-10
Число итераций	2 687	20 988	9 523	6 730
Значение невязки	1.5e-3	5.5e-7	7.5e-5	7.8e-5
Оценка константы Гёльдера m	12 988	500	32 477	107 875
Время (мин.)	2	9	6	5

С одной стороны, ошибка в пятом знаке гарантирует, что модуль разности $\mathbf{r}(t_0 + T) - \mathbf{r}_f$ не превосходит одного процента от астрономической единицы длины, что можно считать достаточным. С другой стороны, не наблюдается устойчивой зависимости между выбранной точностью ε и точностью полученного решения. При сравнимом числе итераций оценка константы Гёльдера может отличаться на порядки. Это заставляет усомниться в выполнении условий теоремы 1.4 из первой главы, гарантирующей сходимости итерационного процесса к глобальному экстремуму.

Наконец, при аналогичном числе итераций заметно значительное уменьшение ошибки при использовании ускоренной модификации алгоритма, при аналогичном числе итераций.

3. Задача оптимального перелета с квадратичным функционалом

Глава устроена следующим образом. Сначала описана формальная постановка задачи, затем с помощью принципа максимума Понтрягина задача оптимального управления сведена к краевой задаче, которая в свою очередь сводится к оптимизационной. В конце главы обсуждаются результаты численных экспериментов.

3.1. Постановка задачи

Рассматривается КА, оборудованный двигателем малой тяги, в центральном гравитационном поле одного массивного тела. Тогда уравнения движения КА принимают следующий вид:

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v}, \\ \dot{\mathbf{v}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a}. \end{cases} \quad (6)$$

Реактивное скорение \mathbf{a} обеспечивается за счет истечения рабочего тела:

$$\mathbf{a} = \frac{\mathbf{F}_T}{m(t)} = \frac{\dot{m}u}{m(t)}\mathbf{e},$$

где \mathbf{F}_T – сила тяги двигателя, u – скорость истечения рабочего тела, \mathbf{e} – единичный вектор тяги. Предположим, что кинетическая энергия, уносимая струей в единицу времени, ограничена величиной N_{max} , с поправкой на КПД двигателя η .

$$\frac{\dot{m}u^2}{2} \leq \eta N_{max}.$$

Выразим скорость истечения через реактивное ускорение:

$$\frac{\mathbf{a}^2 (m_0 - m_r)^2}{2} \leq \eta N_{max}.$$

Проинтегрируем неравенство на временном промежутке $[t_0, t_f]$:

$$\frac{1}{m_0 - m} - \frac{1}{m_0} = \int_{t_0}^{t_f} \frac{\mathbf{a}^2(\tau)}{2\eta N_{max}} d\tau \geq \int_{t_0}^{t_f} \frac{\mathbf{a}^2(\tau)}{2\eta N} d\tau.$$

При использовании всей мощности двигателя неравенство переходит в равенство и достигается минимальный расход топлива, то есть необходимо минимизировать функционал

$$J = \frac{1}{2} \int_{t_0}^{t_f} \frac{\mathbf{a}^2(t)}{\eta N_{max}} dt \rightarrow \min.$$

В предположении постоянства η и N_{max} , вынося константы из-под знака интеграла, имеем следующий оптимизируемый функционал:

$$J = \frac{1}{2} \int_{t_0}^{t_f} \mathbf{a}^2(t) dt \rightarrow \min. \quad (7)$$

Поставим граничные условия задачи встречи (сопровождения):

$$\left\{ \begin{array}{l} \mathbf{r}(t_0) = \mathbf{r}_0, \\ \mathbf{v}(t_0) = \mathbf{v}_0, \\ \mathbf{r}(t_f) = \mathbf{r}_f, \\ \mathbf{v}(t_f) = \mathbf{v}_f. \end{array} \right. \quad (8)$$

Выражения (6), (7) и (8) составляют задачу оптимального управления, где управление \mathbf{a} предполагается неограниченным по величине. Более реалистичной и более сложной задачей представляется задача с постоянной скоростью истечения.

3.2. Принцип максимума

Непрямые методы решения задачи оптимального перелета обычно опираются на принцип максимума Понтрягина.

Фазовому вектору

$$\mathbf{x} = (\mathbf{r}, \mathbf{v}), \mathbf{x} \in \mathbb{R}^6$$

соответствует вектор сопряженных переменных

$$\mathbf{p} = (\mathbf{p}_r, \mathbf{p}_v), \mathbf{p} \in \mathbb{R}^6.$$

В качестве управления можно взять реактивное ускорение, с областью допустимых управлений

$$\mathbf{a} = (0 \leq a < \infty, -\pi/2 \leq \alpha \leq \pi/2, 0 \leq \beta \leq 2\pi), a \in \mathbb{R}^3,$$

где a – модуль ускорения, а углы α и β определяют направление единичного вектора тяги \mathbf{e} . Тогда гамильтониан записывается в виде

$$H = -\frac{\mathbf{a}^2}{2} + \mathbf{p}_r \mathbf{v} + \mathbf{p}_v \left(-\frac{\mu}{r^3} \mathbf{r} + \mathbf{a} \right), \quad (9)$$

и уравнение для эволюции вектора сопряженных имеет вид

$$\dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{x}}.$$

Как легко видеть, управление, доставляющее максимум гамильтониану, есть

$$\mathbf{a} = \mathbf{p}_v.$$

Дифференцируя максимизированный гамильтониан по вектору \mathbf{x} ,

получаем краевую задачу с дифференциальными уравнениями

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{p}_v \\ \dot{\mathbf{p}}_r = -\mathbf{p}_v \\ \dot{\mathbf{p}}_v = -\frac{1}{r^3} \left(3\frac{\mathbf{r} \cdot \mathbf{p}_v}{r^2} - \mathbf{p}_v \right) \end{cases} \quad (10)$$

и краевыми условиями

$$\begin{cases} \mathbf{r}(t_0) = \mathbf{r}_0, \\ \mathbf{v}(t_0) = \mathbf{v}_0, \\ \mathbf{r}(t_f) = \mathbf{r}_f, \\ \mathbf{v}(t_f) = \mathbf{v}_f. \end{cases} \quad (11)$$

Введем вектор невязки \mathbf{f} как

$$\mathbf{f} = \begin{pmatrix} \mathbf{r}(t_f) - \mathbf{r}_t \\ \mathbf{v}(t_f) - \mathbf{v}_t \end{pmatrix}$$

Тогда исходную задачу можно переформулировать как задачу конечномерной оптимизации

$$\|\mathbf{f}\| \rightarrow \min, \quad (12)$$

где функция \mathbf{f} понимается как функция от вектора сопряженных переменных в начальный момент времени и вычисляется интегрированием системы дифференциальных уравнений для фазового вектора и вектора сопряженных переменных.

3.3. Численные эксперименты

Рассматривалась плоская задача сопровождения с квадратичным функционалом. В качестве начальной точки был выбран перигелий земной орбиты, в качестве конечной – афелий орбиты Марса. Начальная и конечная скорость КА были выбраны как скорости планет в соответствующих точках. Использовались безразмерные переменные. Время полета – 499 дней. В качестве алгоритма интегрирования уравнений движения использовался встроенный в модуль `scipy` метод Рунге-Кутты четвертого порядка ($\text{atol} = 10^{-10}$, $\text{rtol} = 10^{-10}$). Вычисления проводились на компьютере Intel Xeon CPU E3-1275 v5 @ 3.60GHz, имеющим 4 ядра, что позволяет распараллелить на максимум 8 потоков.

Сперва был применен параллельный геометрический метод. Было замечено, что при определенных значениях вектора сопряженных переменных в начальный момент времени КА аппарат подлетал слишком близко к Солнцу, что препятствовало численному интегрированию ввиду быстрого роста ошибок округления. Тогда была введена модификация, останавливающая процесс интегрирования при слишком маленьком или слишком большом значении модуля радиус-вектора. В таком случае значение выдаваемой невязки принималось равным заранее определенному числу. Параметры алгоритма принимались следующими: $r = 2$, $\varepsilon = 10^{-6}$. Число разверток и, соответственно, ядер, на которых решалась задача равнялось 7. Алгоритм не сошелся за 100 000 итераций.

Тогда с помощью метода продолжения по параметру [7], было найдено точное решение и запущен алгоритм со стороной гиперкуба 10^{-6} и центром в точном решении. Точность была взята как $\varepsilon = 10^{-6}$. Алгоритм не сошелся за 200 000 итераций. Полученное значение функционала было отлично от нуля в четвертом знаке, каждый из процессов при завершении работы алгоритма имел оценку константы Гёльдера порядка 10^5 . При увеличении

стороны куба до 10^{-5} алгоритм демонстрировал аналогичные результаты, только полученное значение функционала имело ошибку уже в третьем знаке.

Полученное решение представляется неудовлетворительным. Алгоритм оценил константу Гёльдера как величину порядка 10^5 : в силу теоремы 2.2 получаем, что оцениваемая константа Липшица имеет порядок 10^4 . Однако было обнаружено, что в окрестности точного решения функционал очень чувствителен и имеет константу Липшица по крайней мере 10^6 . То есть геометрический алгоритм явно недооценил константу Гёльдера исследуемой задачи и условия теоремы 2.4 нельзя считать выполненными.

Применение ускоренной модификации вылилось в еще большую недооценку константы: например, при значении $\varepsilon = 10^{-3}$ алгоритм завершал свою работу за сотни итераций, находя лишь те точки, в которых выдавалось фиксированное значение невязки, так как КА близко подлетал к Солнцу или, наоборот, слишком отдалялся от центра.

Заключение

В данной работе были реализованы и применены к решению двух задач динамики космического полета алгоритмы глобальной оптимизации, использующие кривые, заполняющие пространство.

Задача оптимального перелета с ограниченной мощностью с помощью принципа максимума Понтрягина была сведена к краевой задаче, а затем введением вектора невязок – к оптимизационной. Аналогично введение вектора невязок позволяет переформулировать краевую задачу Ламберту как задачу конечномерной оптимизации. Использование кривых, заполняющих пространство, обеспечивает редукцию многомерной оптимизационной задачи к одномерной, которая затем решается с помощью геометрического алгоритма. Были реализована ускоренная модификация алгоритма. Алгоритм был распараллелен. Затем были проведены численные эксперименты.

В первой задаче алгоритм позволяет найти значение начального вектора скорости, при котором невязка достаточно мала. Однако время работы слишком большое и не гарантирована сходимость к глобальному минимуму, так как при малом числе итераций оцениваемое значение константы Гёльдера одномерной функции меньше истинного, а значит, условие теоремы о сходимости не выполнено. При решении задачи с квадратичным функционалом возникает та же проблема. Явная недооценка чувствительности задачи к начальным условиям лишает метода одного из его главных достоинств – развитой теории сходимости.

При малом числе итераций, занимающем разумное время, алгоритм не успевает заметить высокую чувствительность задачи. Использование ускоренной модификации алгоритма (по крайней мере на начальном этапе работы алгоритма) в силу этих причин представляется необоснованным. Можно ожидать, что при достаточно большом числе итераций оценка константы Гёльдера станет реалистичной. Однако в таком случае время

работы алгоритма становится слишком большим по сравнению с другими, более эффективными методами [6], [7].

Еще одним принципиальным недостатком метода является ограниченность числа разбиения M . При решении четырехмерной задачи на 64-битном компьютере максимальное значение M равно 16, и найдутся точки, отстоящие от аппроксимации кривой Пеано на 2^{-M-1} , то есть на величину порядка 10^{-5} . Это существенно при больших значениях константы Липшица. Данную проблему можно устранить двумя способами. Во-первых, можно использовать числовую переменную с плавающей точкой `float`, которая реализована на 128, а не 64 битах. Во-вторых, можно запустить алгоритм в гиперкубе с центром в начальном приближении и стороной меньше единицы. Первый способ незначительно улучшает ситуацию, а второй способ требует хорошего начального приближения.

Сведение многомерной задачи к одномерной еще сильнее ухудшает функционал, который изначально обладает большой константой Липшица. Принципиальное ограничение на число M не позволяет добиваться достаточной точности решения задачи. Тем самым алгоритм, основанный на редукции многомерной задаче к одномерной посредством кривых, заполняющих пространство, не позволяет эффективно решить две простейшие в своих классах задачи: задачу Ламберта и задачу оптимизации перелёта с двигателем ограниченной мощности.

Список литературы

- [1] Myatt, D., Vecerra, V., Nasuto, S., Bishop, J.: Advanced Global Optimisation Tools for Mission Analysis and Design. Tech. Rep. 03-4101a, European Space Agency, the Advanced Concepts Team, 2004
- [2] Электрические ракетные двигатели космических аппаратов и их влияние на радиосистемы космической связи / Н.А. Важенин, В.А. Обухов, А.П. Плохих [и др.]. М.: ФИЗМАТЛИТ, 2012.
- [3] Lancaster E.R. and Blanchard R.C. A unified form of Lambert's theorem // NASA technical note TN D-5368, 1969, pp.1-20.
- [4] Battin R.H. An Introduction to the Mathematics and Methods of Astrodynamics. Reston, Virginia: AIAA Education Series, 1999. 796 p.
- [5] Shen, H., Tsiotras, P. Using Battin's method to obtain multiple-revolution Lambert's solutions. 2003.
- [6] Izzo, D.: Revisiting lambert's problem. Celestial Mechanics and Dynamical Astronomy , 2014.
- [7] Petukhov V.G. Optimization of Interplanetary Trajectories for Spacecraft with Ideally Regulated Engines Using the Continuation Method. – Cosmic Research, 2008, Vol. 46, No. 3, pp.219 – 232.
- [8] Simon, D.: Evolutionary Optimization Algorithms, 1nd edn. John Wiley and Sons, Inc., 2013.
- [9] Izzo D., Rucinski M., Biscani F. The generalized Island Model. Parallel Architectures and Bioinspired Algorithms. – Studies in Computational Intelligence, 2012, Vol. 415, pp. 151-169.

- [10] Stiefel, E.L., Scheifele, G. Linear and Regular Celestial Mechanics. Grundlehren der mathematischen Wissenschaften No. 174. Berlin Heidelberg: Springer-Verlag Berlin Heidelberg, 1971.
- [11] Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, USA (2006)
- [12] Torn, A. and Zilinskas, A. 'Global Optimization', Lecture Notes in Computer Science 350, Springer-Verlag, pp. 255, 1989.
- [13] Torn, A., M. Ali and S. Viitanen, 'Stochastic global optimization: Problem classes and solution techniques', Journal of Global Optimization 14, pp. 437-447, 1999.
- [14] Стронгин Р.Г. Численные методы в многоэкстремальных задачах. – М.: Наука, 1978. 240 с.
- [15] Strongin, R.G., Sergeyev, Ya.D.: Global Optimization with Non-convex Constraints: Sequential and Parallel Algorithms. Kluwer, Dordrecht (2000)
- [16] M. Bader, Space-Filling Curves. An Introduction with Applications in Scientific Computing. Springer-Verlag Berlin Heidelberg, 2013.
- [17] Sergeyev Y.D., Strongin R.G., Lera D. Introduction to Global Optimization Exploiting Space-Filling Curves. – Springer Briefs in Optimization. Springer Verlag New-York, 2013. 125 p.
- [18] Баркалов К.А., Сидоров С.В., Рябов В.В. Параллельные вычисления в задачах многоэкстремальной оптимизации. – Вестник Нижегородского университета им. Н.И. Лобачевского, 2009, №6 (1), с. 171-177.
- [19] G. Peano, "Sur une courbe, qui remplit toute une aire plane" Math. Ann., 36(1):157–160, 1890

- [20] D. Hilbert, "Uber die stetige Abbildung einer Linie auf ein Flächenstück" *Math. Annln.*, 38, 459-460, 1891
- [21] Sagan, H.: *Space-Filling Curves*. Springer, New York (1994)
- [22] К. Е. Бауман, "Коэффициент растяжения кривой Пеано-Гильберта"
Матем. заметки, том 80, No 5, стр. 643-656, 2006
- [23] К. Е. Бауман, "Односторонние кривые Пеано фрактального рода 9",
Труды МИАН, т. 275, 2011
- [24] Пиявский С. А. Один алгоритм отыскания абсолютного экстремума функций, *Журнал вычислительной математики и математической физики*, т. 12, No4 (1972), стр. 885—896.
- [25] Gourdin, E., Jaumard, B., Ellaia, R.: Global optimization of Holder functions. *J. Global Optim.* 8, 323–348 (1996)